# Hackathon Project Phases Template

**Project Title:**

**Audio2Art: Transforming Voice Prompts into Visual Creations using Transformers**

**Team Name:**

VocalVisions

**Team Members:**

* Khushi Bung
* Egala Srinidhi Reddy
* Surla Ravali
* Pallakonda Nandhika
* Karnati Sharvani

## Phase-1: Brainstorming & Ideation

**Objective:**

Develop an AI-powered tool that converts voice prompts into visually compelling images using transformer models.

**Key Points:**

1. **Problem Statement:**

* Users struggle to find intuitive ways to generate artwork using just their voice.
* Existing AI art generators require text inputs, limiting accessibility and creativity.

1. **Proposed Solution:**

* An AI-powered application using voice recognition and transformers to create stunning artwork from voice prompts.
* Real-time rendering and customization options for generated images.

1. **Target Users:**

* Artists looking for inspiration
* Content creators needing quick concept art.
* People with disabilities who prefer voice-based interaction.

1. **Expected Outcome:**

* A functional AI tool that takes voice prompts and generates high-quality images using transformers.

## Phase-2: Requirement Analysis

**Objective:**

Define the technical and functional requirements for Audio2Art.

**Key Points:**

1. **Technical Requirements:**

* **Programming Language**: Python
* **Backend:** speechrecognition(googles’ speech recognition service) for voice recognition, Stable Diffusion for image generation.
* **Frontend**: Streamlit Web Framework
* **Database**: Not required initially (API-based queries)

1. **Functional Requirements:**

* Ability to capture voice input and convert it to text.
* Generate images from text using transformer-based models
* Provide users with customization options (styles, color themes, resolution).

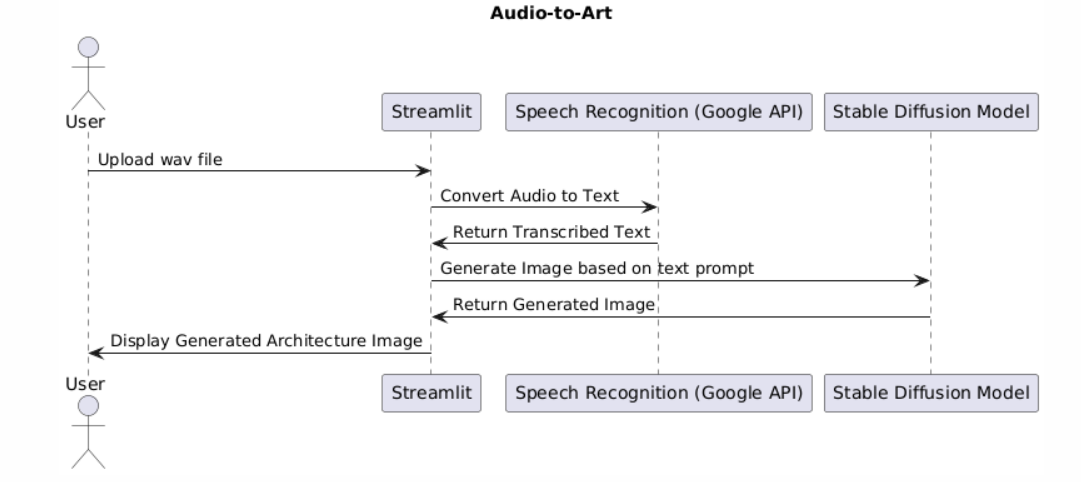
**Constraints & Challenges:**

* Ensuring accurate voice-to-text conversion.
* Optimizing image generation for real-time feedback.

## Phase-3: Project Design

**Objective:**

Develop the architecture and user flow of the application.



**Key Points:**

1. **System Architecture:**

* User speaks a voice prompt into the UI.
* The voice input is processed using speechrecognition.
* Text is passed to a transformer-based image generator.
* Diffusers used to generate images.
* The AI model generates an image and displays it on the frontend.

1. **User Flow:**

* Step 1: User speaks a prompt (e.g., "A futuristic city at sunset").
* Step 2: Voice-to-text conversion occurs.
* Step 3: The text prompt is processed for image generation.
* Step 4: The final image is displayed, with customization options.

1. **UI/UX Considerations:**

* Minimalist, user-friendly interface for seamless navigation.
* Theme selection for different art styles.

## Phase-4: Project Planning (Agile Methodologies)

**Objective:**

Break down development tasks for efficient completion.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Assigned To** | **Dependencies** | **Expected**  **Outcome** |
| Sprint 1 | Environment Setup & API Integration | 🔴 High | 6 hours  (Day 1) | End of Day  1 | Nandhika &  Sharvani | API Keys,  Python setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡  Medium | 2 hours  (Day 1) | End of Day  1 | Ravali | API response format finalized | Basic UI with input fields |
| Sprint 2 | Voice-to-Text  Integration | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Khushi & Srinidhi | Whisper API  Setup | Accurate speech recognition |
| Sprint 2 | Image Generation & Customization | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Khushi &  Srinidhi | Text input Processing ready | Artwork creation from voice input |
| Sprint 3 | Testing & UI  Enhancements | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | Khushi & Srinidhi | API response, UI layout completed | Responsive UI, better user experience |
| Sprint 3 | Final Presentation  & Deployment | 🟢 Low | 1 hour  (Day 2) | End of Day  2 | Entire Team | Working prototype | Demo-ready project |

**Sprint Planning with Priorities**

**Sprint 1 – Setup & Integration (Day 1)**

**(**🔴 **High Priority)** Set up the **environment** & install dependencies.

**(**🔴 **High Priority)** Integrate speechrecognition & Stable Diffusion APIs

**(**🟡 **Medium Priority)** Build a **basic UI with input fields**.

**Sprint 2 – Core Features & Debugging (Day 2)**

**(**🔴 **High Priority)** Implement voice-to-text and image generation functionalities.

**(**🔴 **High Priority)** Debug API issues & handle **errors in queries**.

**Sprint 3 – Testing, Enhancements & Submission (Day 2)**

**(**🟡 **Medium Priority)** Test API responses, refine UI, & fix UI bugs.

**(**🟢 **Low Priority)** Final **demo preparation & deployment**.

## Phase-5: Project Development

**Objective:**

Implement core features of the Audio2Art.

**Key Points:**

1. **Technology Stack Used:**

* **Frontend:** Streamlit
* **Backend:** speechrecognition & Stable Diffusion
* **Programming Language:** Python

1. **Development Process:**

* Implement API authentication for voice recognition and image generation.
* Develop logic for real-time customization of generated artwork.
* Optimize search queries for performance and relevance.

1. **Challenges & Fixes:**

* **Challenge:** Delayed API response times.

**Fix:** Implement caching for frequently used prompts.

* **Challenge:** Handling complex prompts effectively.

**Fix:** Use prompt engineering techniques to optimize results.

## Phase-6: Functional & Performance Testing

**Objective:**

Ensure that the Audio2Art App works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional  Testing | Generate an image from by uploading a wav file | Successfully creates an image | ✅ Passed | Sharvani |
| TC-002 | Functional  Testing | Process multiple overlapping voice inputs | Handles input accurately | ❌ Failed | Ravali |
| TC-003 | Performance  Testing | API response time under  3mins | fast real-time rendering | ⚠ Needs Optimization | Khushi |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect API responses. | Data accuracy should be improved. | ✅ Fixed | Srinidhi |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | ❌ Failed - UI broken on mobile | Nandhika |
| TC-006 | Deployment  Testing | Host the app using  Streamlit Sharing | App should be accessible online. | 🚀 Deployed | Khushi |

## Final Submission

1. Project Report Based on the templates
2. Demo Video (3-5 Minutes)
3. GitHub/Code Repository Link
4. Presentation